Chapter 14. Pointer Usage

Pointers are used within PDS labels to indicate the relative locations of objects in the same file and to reference external files. Pointer statements begin with a caret ("^") and the name of a PDS object or element. The value part of the pointer statement indicates the location of the referenced information.

14.1   Types of Pointers

Pointer statements fall into three main categories: data location pointers, include pointers, and related information pointers.

14.1.1 Data Location Pointers (Data Object Pointers)

The most common use of pointers occurs in PDS labels, for linking object descriptions to the actual data. The syntax of these pointers depends on whether the label is attached or detached from the data it describes. There are five forms for the value fields, as shown in these examples:

```
(1)    ^IMAGE             = 12
(2)    ^IMAGE             = 600 <BYTES>
(3)    ^INDEX_TABLE       = "INDEX.TAB"
(4)    ^SERIES            = ("C100306.DAT", 2)
(5)    ^SERIES            = ("C100306.DAT", 700 <BYTES>)
```

Examples (1) and (2) are pointers in attached labels. This type of pointer allows reading software to scan the label for the appropriate pointer and then skip right to the data at its location elsewhere in the file. In the first case, the data begin at record 12 of the labeled file. In the second, the data begin at byte 600.

External data files are referenced in examples (3), (4) and (5). Since these pointers occur in detached labels, they must identify a file name and (optional) offset. In example (3), the data begin at record 1 of the data file "INDEX.TAB" (i.e., no explicit offset is taken as an offset of "1"). In example (4), the data begin at record 2 of the data file, "C100306.DAT", whereas in example (5), the data begin at byte 700.

14.1.2 Include Pointers

Another common use of pointers is to reference external files in PDS labels or catalog objects. Files referenced by include pointers are included directly at the location of the pointer statement. These pointers are classified as include-type pointers since they act like the "#include" statements in C program source files. Pointers with the class names of STRUCTURE, CATALOG, and MAP_PROJECTION fall into this category. The included files contain only PDS data object definitions or completed catalog objects. Following are some examples of include pointer statements:

> (1)   ^STRUCTURE = "ENGTAB.FMT"
> (2)   ^STRUCTURE = "IMAGE.FMT"
> (3)   ^CATALOG = "CATALOG.CAT"
> (4)   ^DATA_SET_MAP_PROJECTION = "DSMAPDIM.CAT"

The structure file in example (1) is referenced by a TABLE object. The "ENGTAB.FMT" file contains column object definitions needed to complete the TABLE definition. Some column definitions might be stored in a separate file if, for example, a number of different TABLE objects use the same definitions. Similarly, in example (2) an IMAGE object definition (i.e., all statements beginning with "OBJECT = IMAGE" and ending with "END_OBJECT = IMAGE") is contained in an external file called "IMAGE.FMT".

In example (3), the external file "CATALOG.CAT" is referenced by a VOLUME object in order to provide a full set of catalog information associated with the volume without having to duplicate definitions that already exist in the other file.

In example (4), the external file "DSMAPDIM.CAT" is referenced by an IMAGE_MAP_PROJECTION object to complete the map projection information associated with the image.

14.1.3 Related Information Pointers (Description Pointers)

The third and final use of pointers occurs in PDS labels that reference external files of additional documentation of special use to human readers. These pointers are formed using elements that end in "DESCRIPTION" or "DESC". They reference text files not written in ODL. Note: These pointers are *not* meant to be used to refer to software tools.

For example:

> ^DESCRIPTION = "TRK_2_25.ASC"

In this example, the pointer references an external ASCII document file, TRK_2_25.ASC, which provides a detailed description of the data. Note that in this case the documentation file must have its own PDS label, since the label containing the ^DESCRIPTION pointer describes the contents of a different file.

14.2   Rules for Resolving Pointers

Following are the rules for resolving pointer references to external files (see the *Volume Organization and Naming* chapter in this document for information about physical and logical volume structures):

For a pointer statement in FILE_A:

(1)   Look in the same directory as FILE_A

(2a)   For a single physical volume (no logical volumes), look in the following top level directory:

Pointer	Directory
^STRUCTURE	LABEL
^CATALOG	CATALOG
^DATA_SET_MAP_PROJECTION	CATALOG*
^INDEX_TABLE	INDEX
^DESCRIPTION or ^TEXT	DOCUMENT

(2b)   Within a logical volume, look in the top level subdirectory specified by the LOGICAL_VOLUME_PATH_NAME keyword:

Pointer	LOGICAL_VOLUME_PATH_NAME / Directory
^STRUCTURE	LABEL
^CATALOG	CATALOG
^DATA_SET_MAP_PROJECTION	CATALOG*
^INDEX_TABLE	INDEX
^DESCRIPTION or ^TEXT	DOCUMENT

* Note: For volumes using PDS Version 1 or 2 standards, the MAP_PROJECTION files may be located in the LABEL directory

All pointers to data objects should be resolved in step (1), since these files are always required to be located in the same directory as the label file.